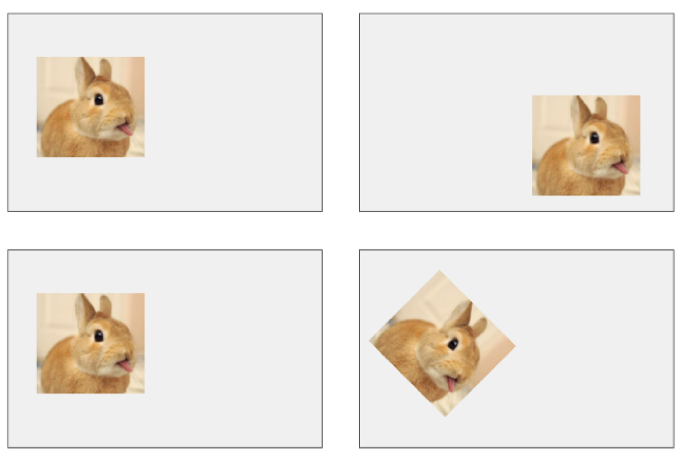
Tensor field networks: 3D rotation-equivariant convolutional neural networks

Google Accelerated Science

Tess Smidt (blondegeek@), Nathaniel Thomas (Stanford), Steve Kearnes (kearnes@), Lusann Yang (lusann@), Kai Kohlhoff (kohlhoff@), Patrick Riley (pfr@)

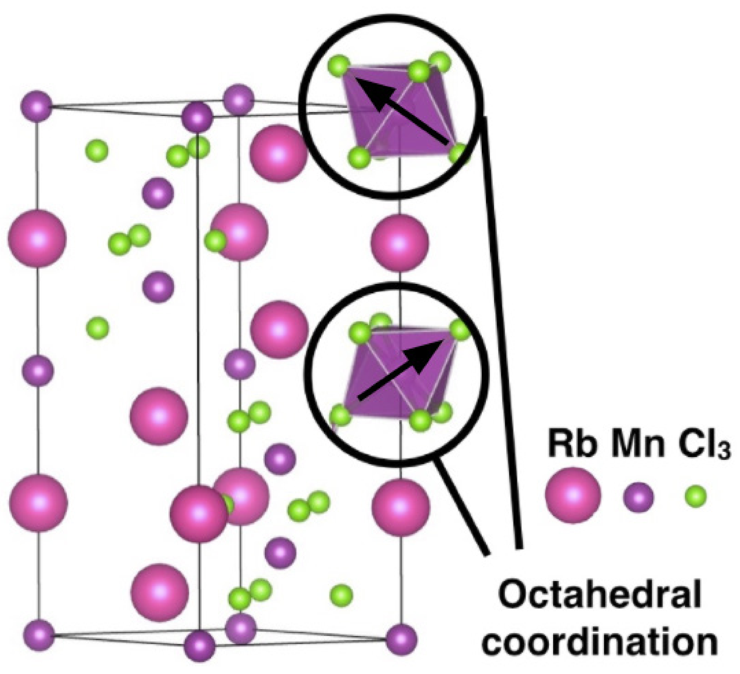
Why 3D rotation-equivariance?



Translation equivariance
Convolutional neural network ✓

Rotation equivariance
~~Data augmentation~~ (x1000 in 3D!)
~~Radial functions~~
Want a network that both preserves geometry and exploits symmetry.

A network with 3d translation- and rotation-equivariance would allow us to identify chemical motifs.

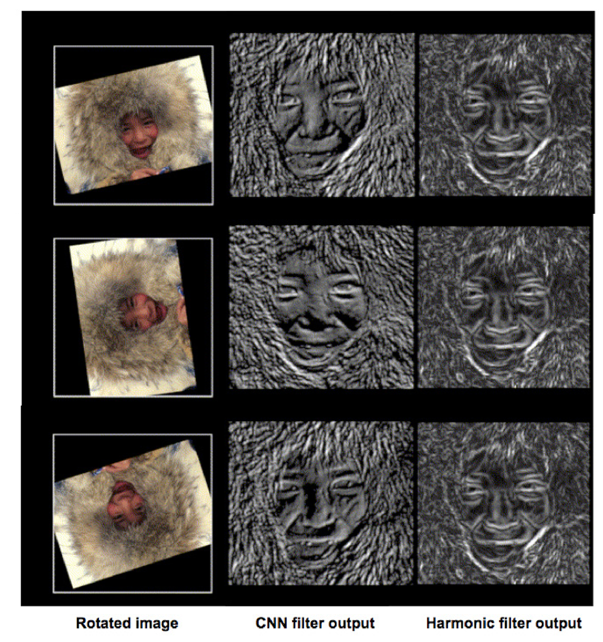


Previous Work

2D Rotation-Equivariance

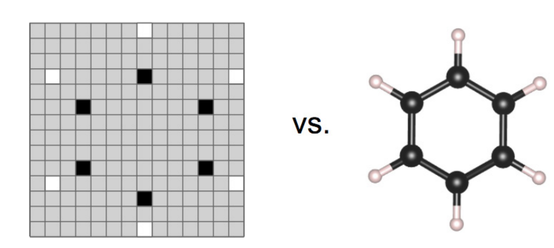
Previous work on 2d rotation-equivariance uses filters based on circular harmonics [1].

To extend to 3d, we use spherical harmonics.

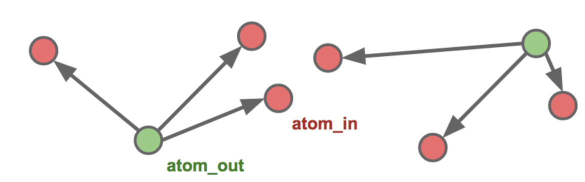


Continuous Convolutions

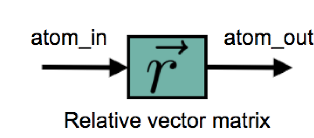
Images that represent atomic systems are both sparse and insufficiently precise.



Instead, We represent atoms as labeled points. These points serve as convolution centers.

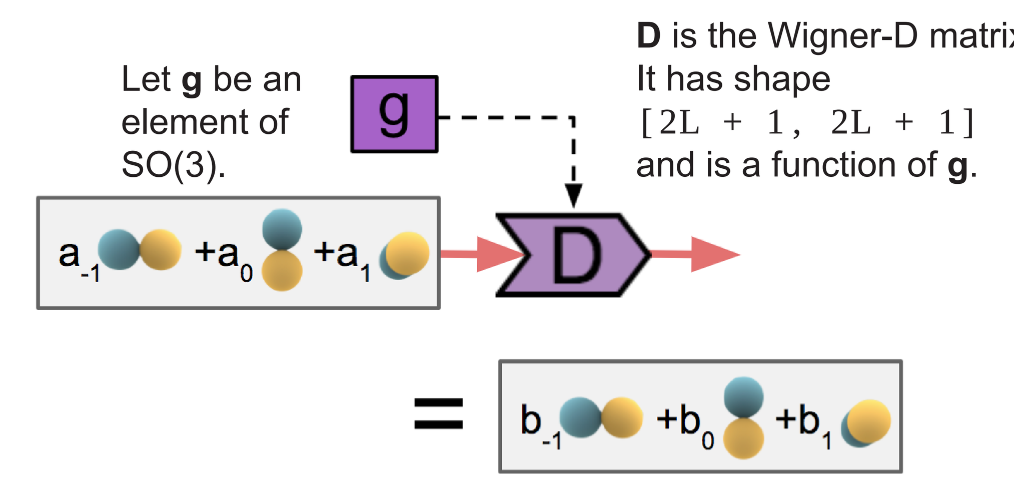


This has been used by other networks operating on atomic structure data [2].

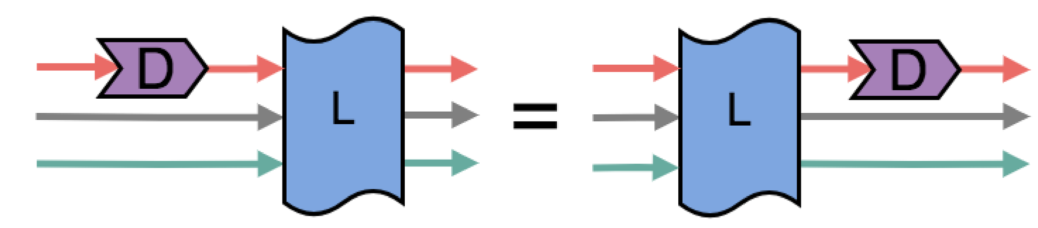


Visual Proof of Rotation-Equivariance

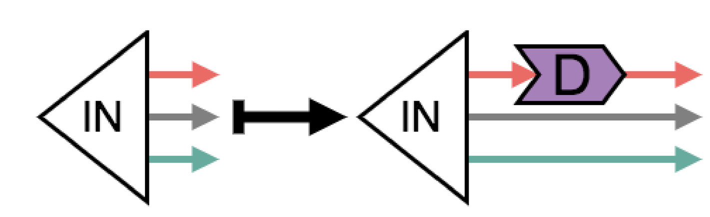
Spherical harmonics of a given L transform together under rotation.



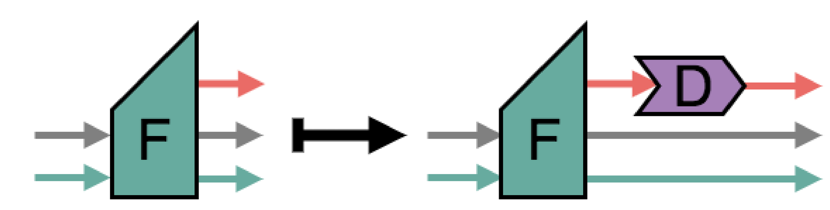
To be rotation-equivariant, the following relationship must hold for each layer:



Each component of the network must be individually rotation-equivariant to guarantee the network rotation equivariance.

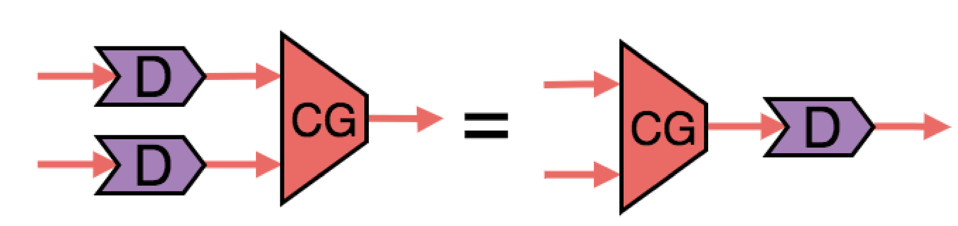


(1)



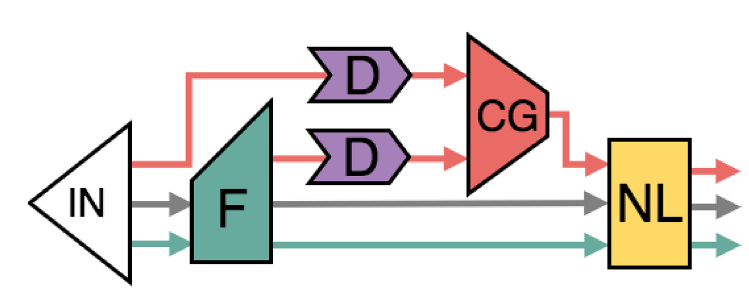
(2)

The key property of the Clebsch-Gordan coefficient tensor:

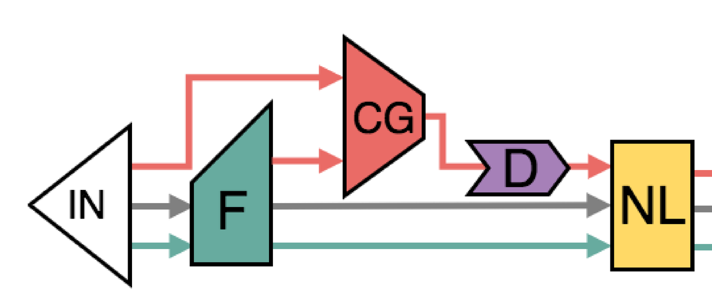


(3)

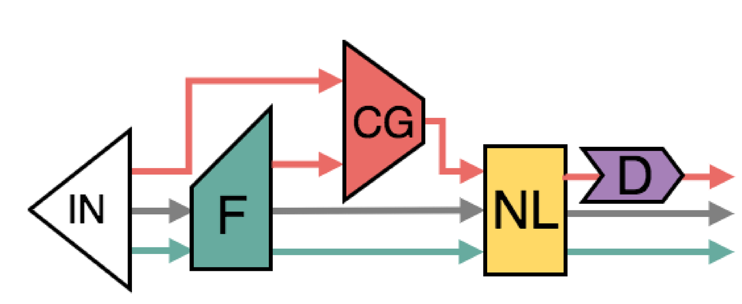
Putting it all together, we can show that the layers are equivariant.



From (1) and (2).

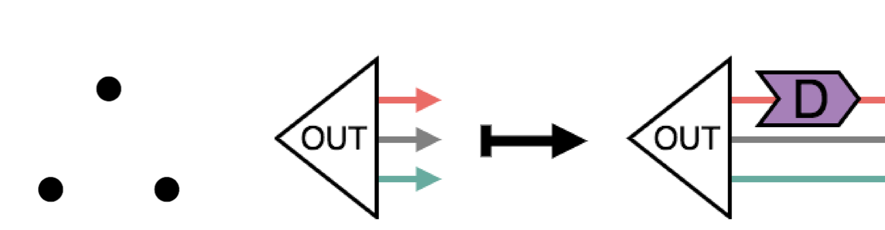


From (3).



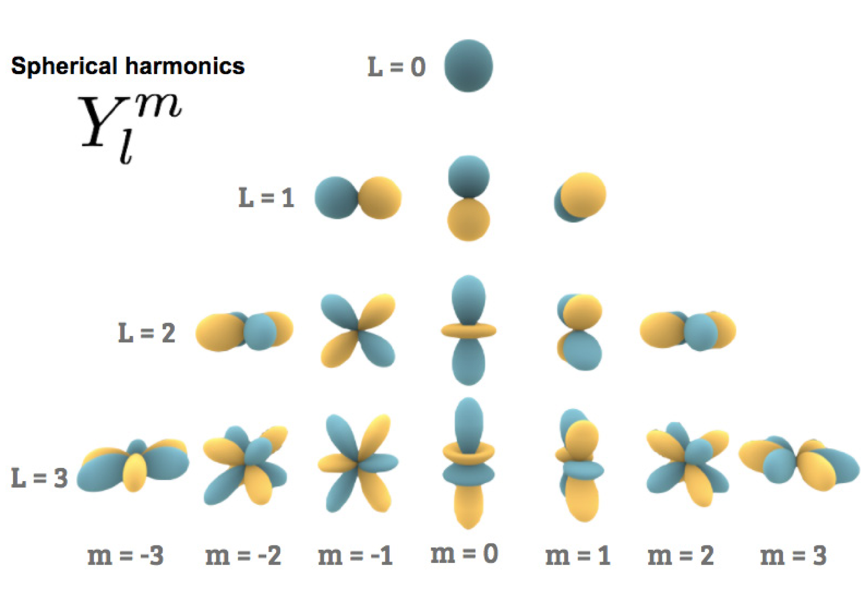
NL is a scalar.

∴



Representation Theory: Spherical Harmonics, Tensors, and Tensor Products

Spherical harmonics are the basis functions of irreducible representations (irrep) of the group of 3D rotations, SO(3). Each spherical harmonic is indexed by a L and a M. For each irrep L, there are 2L + 1 functions and -L<=M<=L.



Tensors are defined by their transformation properties.

Let R be a 3D rotation matrix. Scalar s, vector v, and rank-2 tensor t transform as:

$$s \mapsto s$$
$$v_i \mapsto Rv_i$$
$$t_{ij} \mapsto Rt_{ij}R^T$$

These mathematical objects project onto L irreps in the following way:

$$s : L = 0$$
$$v_i : L = 1$$
$$t_{ij} : L = 0 \oplus 1 \oplus 2$$

Any rank-N tensor can be expressed as a tensor sum of L <= N irreps.

To combine two tensors, one uses tensor products which operate on the projection of the tensor onto Ls in the following way:

$$L_1 \otimes L_2 = \bigoplus_{L=|L_1-L_2|}^{L_1+L_2} L$$

e.g.

$$0 \otimes 1 = 1$$
$$1 \otimes 1 = 0 \oplus 1 \oplus 2$$
$$2 \otimes 1 = 1 \oplus 2 \oplus 3$$

Clebsch-Gordan coefficients are used to combine specific contributions from L1, M1 and L2, M2 to L, M.

Network Components and Operations

This network natively handles tensors defined at 3D points (tensor fields) as input and output.

Each filters and input have an associated L ***and*** M***. These representations must be combined using the Clebsch-Gordan tensor to give the output.***

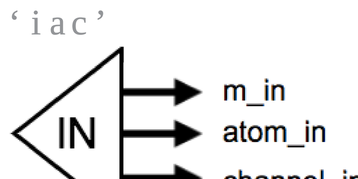
Input features are projected onto L**'s. The input to the network is a dictionary with** L **keys and** `tf.Tensor` **values**

Each tensor is size:
`[2L + 1, num_atoms, num_channels]`

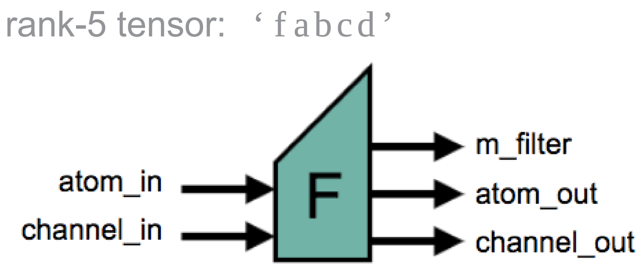
We will use the following indices to express `tf.Tensor` **manipulations using** `tf.einsum` **notation.**

	Shape:
i: m_input	2 L_input + 1
f: m_filter	2 L_filter + 1
o: m_output	2 L_output + 1
a: atom_in	num_atoms
b: atom_out	num_atoms
c: channel_in	
d: channel_out	

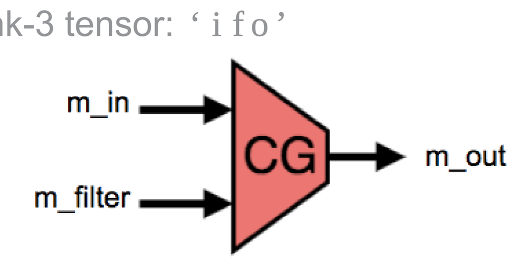
Input
rank-3 tensor: 'iac'



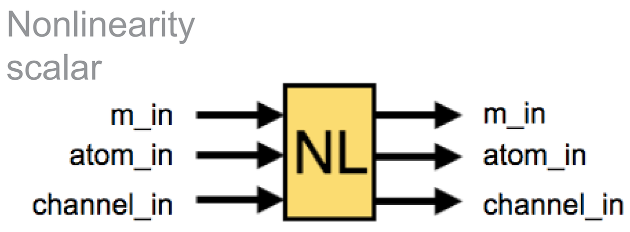
Filter
rank-5 tensor: 'fabcd'

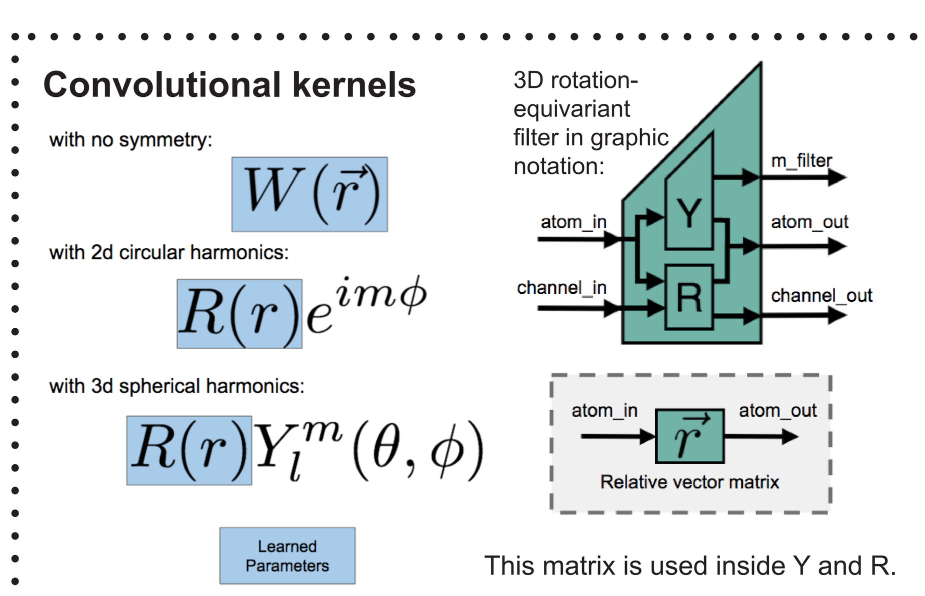


Clebsch-Gordan tensor
rank-3 tensor: 'ifo'

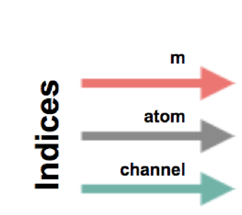


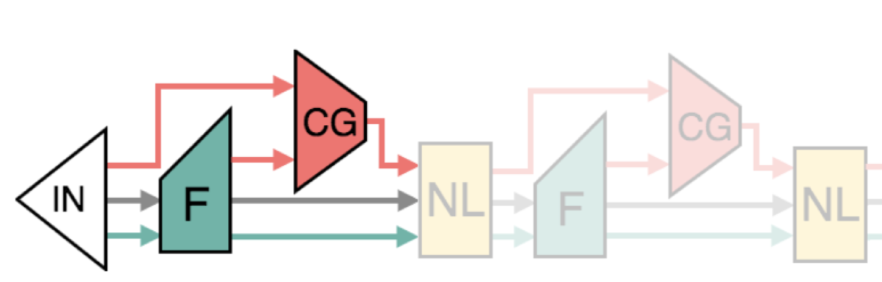
Nonlinearity
scalar





The tensor operations for one layer in `tf.einsum` **notation:**
'iac,fabcd,ifo->obd'





Datasets

We are actively looking for datasets to try!

We are currently training on small molecule property datasets (QM9, MD17) and toy datasets to demonstrate the capabilities of our network.

For toy datasets: we are having the network predict gravitational accelerations of point masses to demonstrate L=1 predictions and moment of inertia tensors to demonstrate L=0+2 predictions.

Got tensors? Got geometry? Talk to us!

References

[1] Worrall, D. E., Garbin, S. J., Turmukhambetov, D. & Brostow, G. J. Harmonic Networks: Deep Translation and Rotation Equivariance. arXiv:1612.04642, (2016).

[2] K. T. Schütt, P.J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K.R. Müller. SchNet: A continuousfilter convolutional neural network for modeling quantum interactions. arXiv:1706.08566, (2017).